สามยอดเขา

เทือกเขาคอร์ดิลเลราคือแนวภูเขาส่วนหนึ่งของเทือกเขาแอนเดสซึ่งทอดยาวข้ามโบลิเวีย ซึ่งประกอบด้วยลำดับของ N ยอดเขา ระบุด้วยหมายเลข 0 ถึง $N-1$ ยอดเขาที่ i ($0 \le i < N$) มี**ความสูง** เท่ากับ $H[i]$ ซึ่งเป็นจำนวนเต็มที่มีค่าอยู่ระหว่าง 1 ถึง $N-1$ รวมหัวท้าย

สำหรับยอดเขา i และ j ที่ $0 \le i < j < N$ เราจะกล่าวว่ายอดเขาคู่ดังกล่าวมี **ระยะห่าง** เท่ากับ $d(i,j) = j - i$

ตามตำนานของชาวอินคาโบราณ สามยอดใด ๆ จะเป็น **กลุ่มเขาในตำนาน** ถ้ามีคุณสมบัติเฉพาะดังนี้: ความสูงของยอดเขาทั้งสาม**ตรง**กับระยะห่างระหว่างคู่ของยอดเขาในกลุ่มนั้น **เมื่อไม่สนใจลำดับ**

กล่าวอย่างเป็นทางการคือ กลุ่มของยอดเขาหมายเลข (i,j,k) จะเป็นกลุ่มเขาในตำนานเมื่อ

- $0 \le i < j < k < N$ และ
- ความสูง $(H[i], H[j], H[k])$ นั้นประกอบด้วยเลขชุดเดียวกับระยะห่างระหว่างแต่ละคู่ $(d(i,j), d(i,k), d(j,k))$ เมื่อไม่สนใจลำดับ ตัวอย่างเช่น สำหรับยอดเขาหมายเลข $0,1,2$ ระยะห่างระหว่างแต่ละคู่เป็น $(1,2,1)$ ดังนั้นถ้าเป็นความสูง $(H[0],H[1],H[2]) = (1,1,2)$, $(H[0],H[1],H[2]) = (1,2,1)$ และ $(H[0],H[1],H[2]) = (2,1,1)$ ล้วนตรงกับระยะห่างดังกล่าว แต่ถ้าเป็นความสูง $(H[0],H[1],H[2]) = (1,2,2)$ จะไม่ตรง

โจทย์ข้อนี้ประกอบด้วยสองตอน โดยแต่ละปัญหาย่อยจะเป็นของ **ตอนที่ I** หรือ **ตอนที่ II** คุณสามารถแก้ปัญหาย่อยตามลำดับอย่างไรก็ได้ กล่าวคือ คุณ**ไม่จำเป็น**ต้องทำตอนที่ I ให้เสร็จทั้งหมดก่อนที่จะทำตอนที่ II

ตอนที่ I

จากรายละเอียดของเทือกเขาที่ให้ งานของคุณคือการนับจำนวนกลุ่มยอดเขาที่เป็นกลุ่มเขาในตำนาน

รายละเอียดการเขียนโปรแกรม

คุณต้องเขียนฟังก์ชันต่อไปนี้

```
long long count_triples(std::vector<int> H)
```

- H: อาร์เรย์ความยาว N แทนความสูงของยอดเขาทั้งหมด
- ฟังก์ชันนี้จะถูกเรียกใช้หนึ่งครั้งสำหรับแต่ละข้อมูลทดสอบ

ฟังก์ชันต้องคืนค่าจำนวนเต็ม T ซึ่งแทนจำนวนกลุ่มเขาในตำนานที่ปรากฏในเทือกเขานี้

เงื่อนไข

- $3 \le N \le 200\,000$
- $1 \le H[i] \le N - 1$ สำหรับแต่ละ i ที่ $0 \le i < N$

ปัญหาย่อย

ตอนที่ I มีคะแนนรวมเท่ากับ 70 คะแนน

ปัญหาย่อย	คะแนน	เงื่อนไขเพิ่มเติม
1	8	$N \le 100$
2	6	$H[i] \le 10$ สำหรับแต่ละ i ที่ $0 \le i < N$
3	10	$N \le 2000$
4	11	ความสูงจะมีค่าไม่ลดลง กล่าวคือ $H[i-1] \le H[i]$ สำหรับแต่ละ i ที่ $1 \le i < N$
5	16	$N \le 50\,000$
6	19	ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

พิจารณาการเรียกดังนี้

```
count_triples([4, 1, 4, 3, 2, 6, 1])
```

เทือกเขานี้มีกลุ่มเขาในตำนาน 3 กลุ่มได้แก่:

- เมื่อ $(i, j, k) = (1, 3, 4)$ ความสูง $(1, 3, 2)$ ตรงกับระยะห่าง $(2, 3, 1)$
- เมื่อ $(i, j, k) = (2, 3, 6)$ ความสูง $(4, 3, 1)$ ตรงกับระยะห่าง $(1, 4, 3)$
- เมื่อ $(i, j, k) = (3, 4, 6)$ ความสูง $(3, 2, 1)$ ตรงกับระยะห่าง $(1, 3, 2)$

ดังนั้นฟังก์ชันต้องคืนค่าเป็น 3

สังเกตว่ากลุ่มของยอดเขาหมายเลข $(0, 2, 4)$ ไม่เป็นกลุ่มเขาในตำนาน เนื่องจากความสูง $(4, 4, 2)$ ไม่ตรงกับระยะห่าง $(2, 4, 2)$

ตอนที่ II

งานของคุณคือการสร้างแนวเทือกเขาที่ประกอบด้วยกลุ่มเขาในตำนานจำนวนต่าง ๆ โดยตอนนี้จะประกอบด้วย 6 ปัญหาย่อยสำหรับการส่ง**เฉพาะผลลัพธ์** (output-only) โดยมีการคิด**คะแนนบางส่วน** (partial scoring)

ในแต่ละปัญหาย่อย คุณจะได้รับจำนวนเต็มบวกสองตัวได้แก่ M และ K และคุณต้องสร้างแนวเทือกเขาที่มียอดเขา**อย่างมาก** M ยอด ถ้าคำตอบของคุณประกอบด้วยกลุ่มเขาในตำนาน**อย่างน้อย** K กลุ่ม คุณจะได้รับคะแนนเต็มในปัญหาย่อย

นั้น ในกรณีอื่นคะแนนของคุณจะถูกคิดตามสัดส่วนของจำนวนกลุ่มเขาในตำนานที่ปรากฏในคำตอบของคุณ

สังเกตว่าคำตอบของคุณจะต้องประกอบด้วยข้อมูลเทือกเขาที่ถูกต้อง กล่าวอย่างชัดเจนคือ สมมติว่าคำตอบของคุณมียอดเขา N ยอด (N ต้องมีค่า $3 \leq N \leq M$) ดังนั้น ความสูงของยอดเขา i ($0 \leq i < N$) กำหนดด้วย $H[i]$ จะต้องมีค่าอยู่ระหว่าง 1 ถึง $N-1$ รวมหัวท้าย

รายละเอียดการเขียนโปรแกรม

มีสองวิธีสำหรับการส่งคำตอบของคุณ ซึ่งคุณสามารถใช้วิธีใดก็ได้ในแต่ละปัญหาย่อย:

- **ไฟล์ผลลัพธ์**
- **การเรียกฟังก์ชัน**

หากต้องการส่งคำตอบด้วย**ไฟล์ผลลัพธ์** ให้สร้างและส่งไฟล์ข้อความในรูปแบบต่อไปนี้:

```
N
H[0]  H[1] ... H[N-1]
```

หากต้องการส่งคำตอบด้วย**การเรียกฟังก์ชัน** คุณต้องเขียนฟังก์ชันต่อไปนี้

```
std::vector<int> construct_range(int M, int K)
```

- M: จำนวนยอดเขาที่มากที่สุด
- K: จำนวนกลุ่มเขาในตำนานที่ต้องการ
- ฟังก์ชันนี้จะถูกเรียกใช้หนึ่งครั้งสำหรับแต่ละปัญหาย่อย

ฟังก์ชันจะต้องคืนค่าเป็นอาร์เรย์ H ความยาว N เป็นตัวแทนของความสูงของยอดเขาทั้งหมด

ปัญหาย่อยและการให้คะแนน

ตอนที่ II มีคะแนนรวมเท่ากับ 30 คะแนน สำหรับแต่ละปัญหาย่อย ค่าของ M และ K จะถูกกำหนดให้ตามตารางต่อไปนี้:

ปัญหาย่อย	คะแนน	M	K
7	5	20	30
8	5	500	2000
9	5	5000	$50\,000$
10	5	$30\,000$	$700\,000$
11	5	$100\,000$	$2\,000\,000$
12	5	$200\,000$	$12\,000\,000$

สำหรับแต่ละปัญหาย่อย ถ้าคำตอบของคุณไม่สอดคล้องกับแนวเทือกเขาที่ถูกต้องตามกำหนด คะแนนของคุณจะเป็น 0 (รายงานว่า `Output isn't correct` ใน CMS)

ในกรณีอื่น ให้ T แทนจำนวนกลุ่มเขาในตำนานในคำตอบของคุณ คะแนนของคุณในปัญหาย่อยนั้นจะเป็น

$$5 \cdot \min\left(1, \frac{T}{K}\right)$$

เกรดเดอร์ตัวอย่าง

ตอนที่ I และ II ใช้โปรแกรมเกรดเดอร์ตัวอย่างเดียวกัน โดยความแตกต่างระหว่างทั้งสองตอนถูกกำหนดโดยบรรทัดแรกของข้อมูลนำเข้า

รูปแบบข้อมูลนำเข้าสำหรับตอนที่ I:

```
1
N
H[0]  H[1] ... H[N-1]
```

รูปแบบข้อมูลส่งออกสำหรับตอนที่ I:

```
T
```

รูปแบบข้อมูลนำเข้าสำหรับตอนที่ II:

```
2
M K
```

รูปแบบข้อมูลส่งออกสำหรับตอนที่ II:

```
N
H[0]  H[1] ... H[N-1]
```

สังเกตว่าข้อมูลส่งออกของเกรดเดอร์ตัวอย่างสอดคล้องกับรูปแบบที่ต้องการสำหรับไฟล์ผลลัพธ์ในตอนที่ II